離散最適化基礎論

ジョブ・スケジューリングのアルゴリズム

第 11 回

ショップ・スケジューリング：機械数が定数

岡本 吉央　(電気通信大学)
okamotoy@uec.ac.jp

2025 年 1 月 7 日

最終更新：2025 年 1 月 6 日　15:15

スケジュール (前半)

1. スケジューリング問題の分類 (10/1)
 ＊ 休み (出張) (10/8)
 ＊ 休み (体育祭) (10/15)
2. 整列による解法 (10/22)
3. 動的計画法 (10/29)
4. NP 困難性と計算量の分類 (11/5)
5. 計算複雑性による問題の分類 (11/12)
6. リスト・スケジューリング (11/19)

スケジュール (後半)

7.	先行制約：基礎	(11/26)
＊	休み (秋ターム試験)	(12/3)
8.	先行制約：多機械	(12/10)
9.	先行制約：他の半順序	(12/17)
10.	ショップ・スケジューリング：基礎	(12/24)
＊	休み (冬季休業)	(12/31)
11.	**ショップ・スケジューリング：機械数が定数**	(1/7)
12.	ショップ・スケジューリング：機械数が可変	(1/14)
13.	近似可能性と近似不可能性	(1/21)
14.	多項式時間近似スキーム	(1/28)
＊	なし	(2/4)

[復習] ジョブと工程

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

記法：O_{ij} = ジョブ J_j を構成する工程で，
機械 M_i で処理するもの

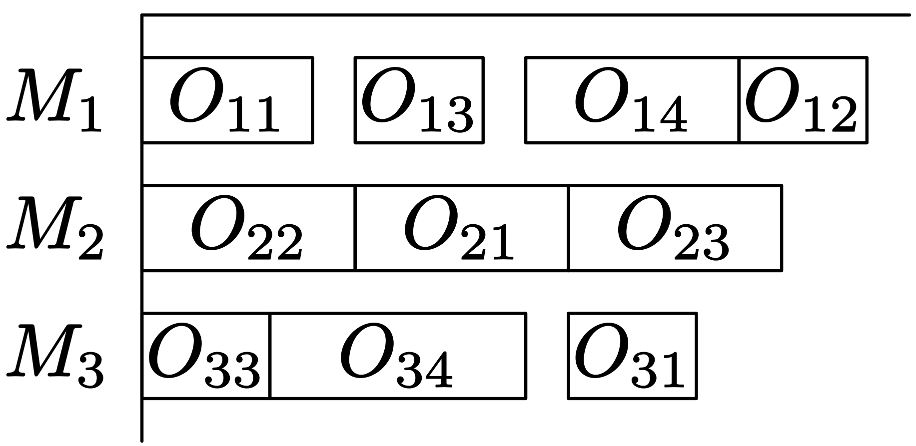
p_{ij} = 工程 O_{ij} の処理時間

注：
1 つのジョブにおける複数の工程を同じ機械で処理する場合も考えられているが，この授業では扱わない

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & \\
M_3 & O_{31} & & O_{33} & O_{34} \\
\hline
\end{array}
$$

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

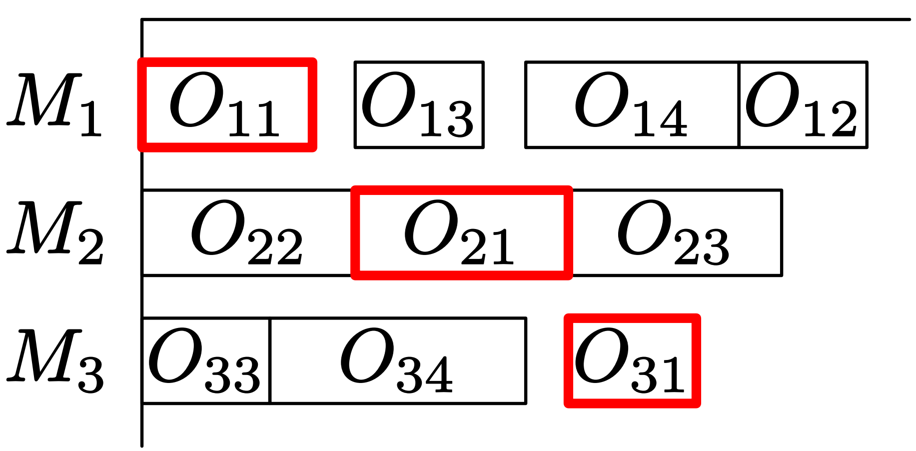


$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & \\
M_3 & O_{31} & & O_{33} & O_{34} \\
\hline
\end{array}
$$

[復習] ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

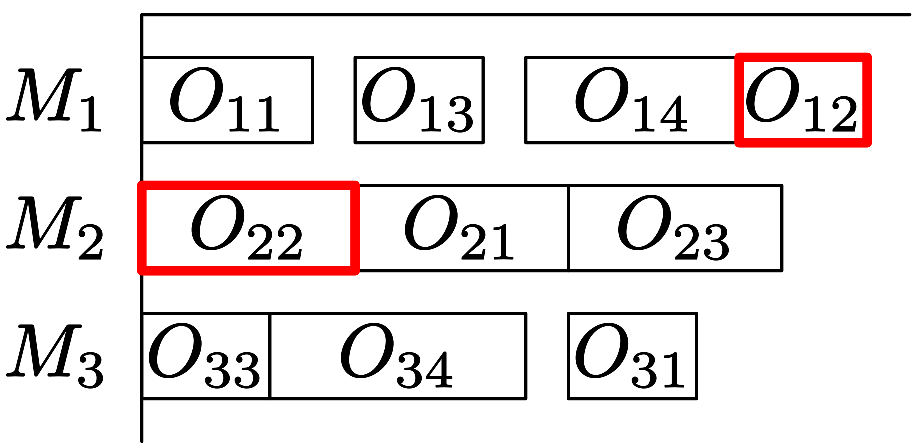


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

[復習] ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

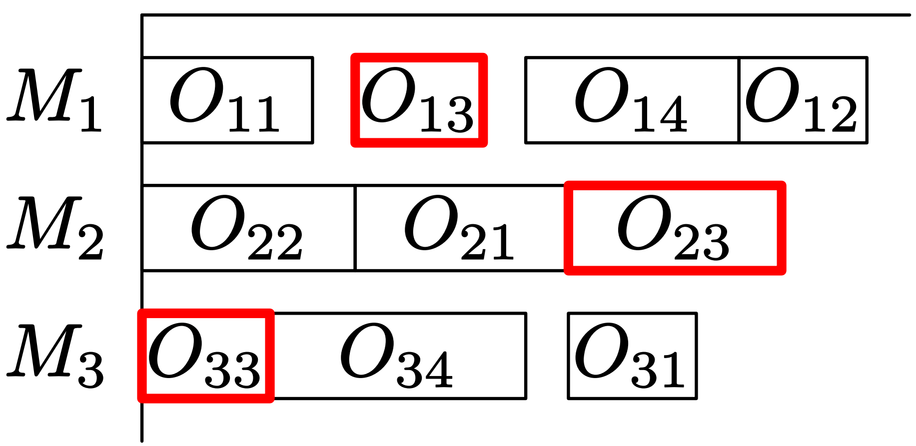


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

[復習] ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

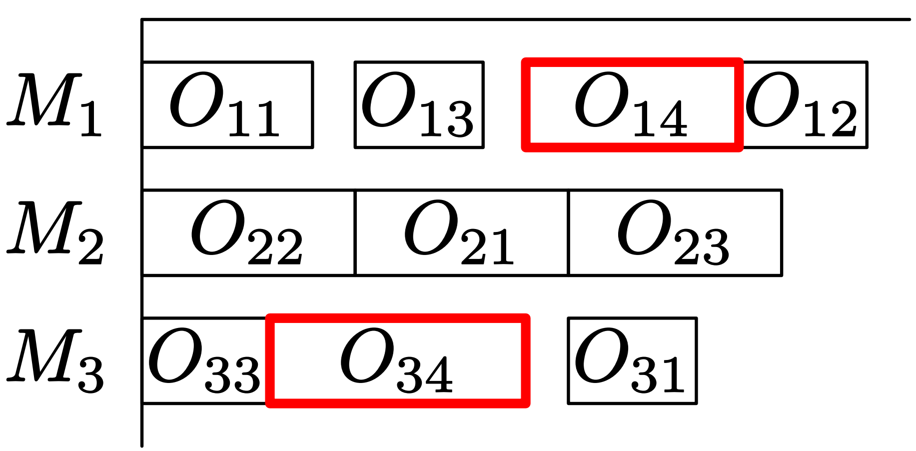
- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない



	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない



$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & \\
M_3 & O_{31} & & O_{33} & O_{34} \\
\hline
\end{array}
$$

次の 3 つの制限・変種をよく考える

オープンショップ・スケジューリング では

- 任意のジョブ J_j と機械 M_i に対して, 工程 O_{ij} がある

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} \\
M_3 & O_{31} & O_{32} & O_{33} & O_{34} \\
\hline
\end{array}
$$

次の 3 つの制限・変種をよく考える

フローショップ・スケジューリング では

- 任意のジョブ J_j と機械 M_i に対して, 工程 O_{ij} がある
- $O_{ij} \to O_{i+1,j}$ という先行制約がある

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_3 & O_{31} & O_{32} & O_{33} & O_{34} \\
\hline
\end{array}
$$

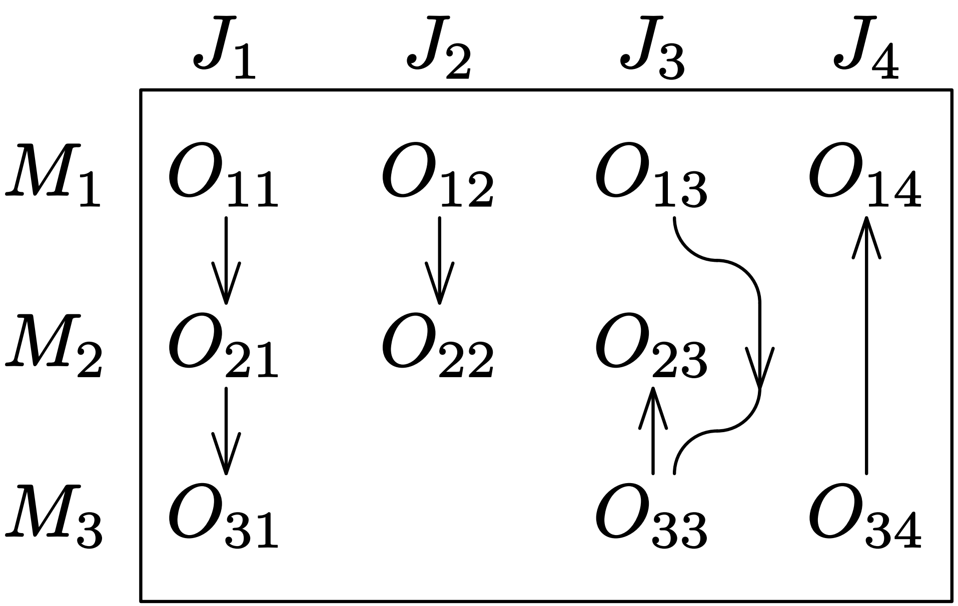
直感：流れ作業

[復習] ジョブショップ・スケジューリング

次の 3 つの制限・変種をよく考える

ジョブショップ・スケジューリング では

- 各ジョブに対して，その工程の間に
  鎖で表される先行制約がある



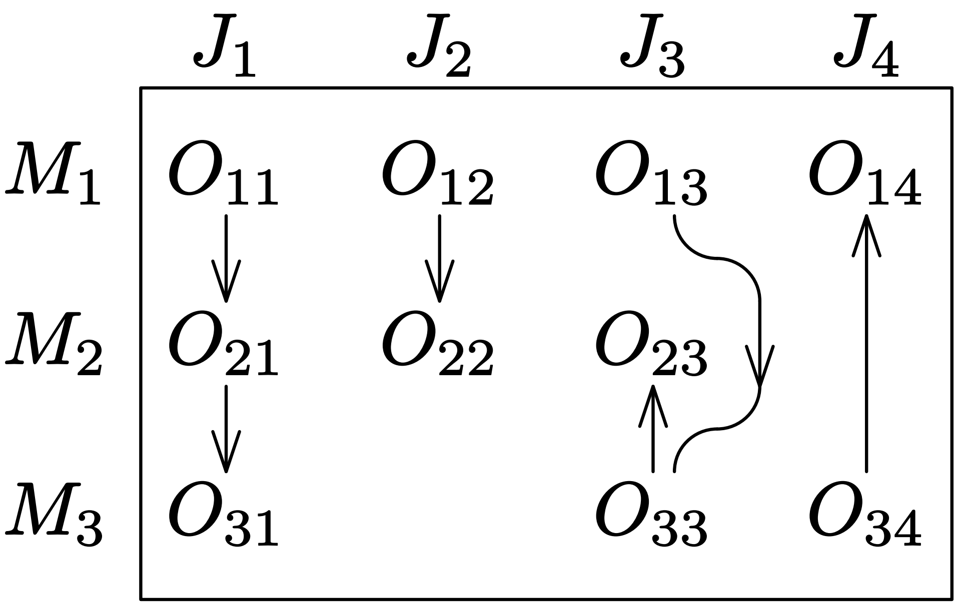
注：フローショップはジョブショップの特別な場合

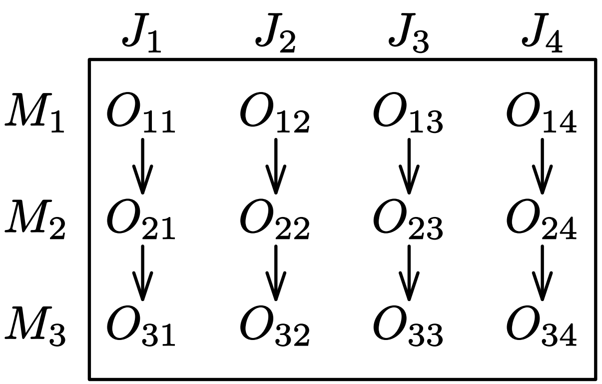
[復習] ジョブショップ・スケジューリング

次の 3 つの制限・変種をよく考える

ジョブショップ・スケジューリング では

- 各ジョブに対して，その工程の間に
  鎖で表される先行制約がある





注：フローショップはジョブショップの特別な場合

[復習] 3 つ組記法

3 つ組記法 $\alpha \mid \beta \mid \gamma$ の α に次を書く ($\alpha =$ 機械の環境)

- O オープンショップ 機械数は入力の一部
- F フローショップ 機械数は入力の一部
- J ジョブショップ 機械数は入力の一部

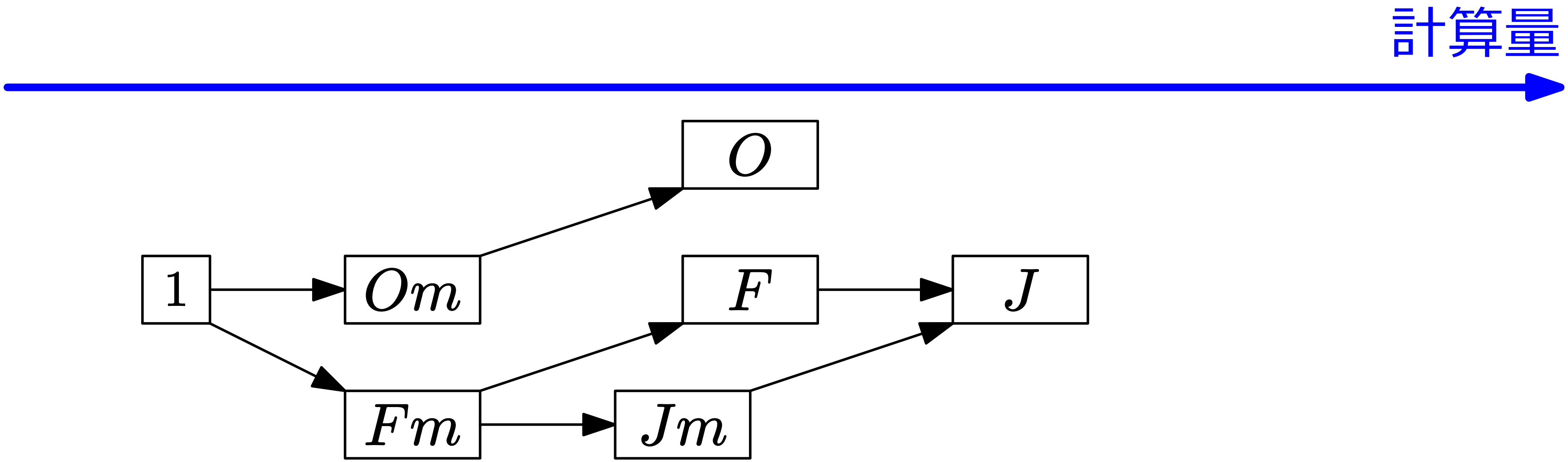
- Om オープンショップ 機械数は m で固定
- Fm フローショップ 機械数は m で固定
- Jm ジョブショップ 機械数は m で固定

[復習] 3 つ組記法

3 つ組記法 $\alpha \mid \beta \mid \gamma$ の α に次を書く ($\alpha =$ 機械の環境)

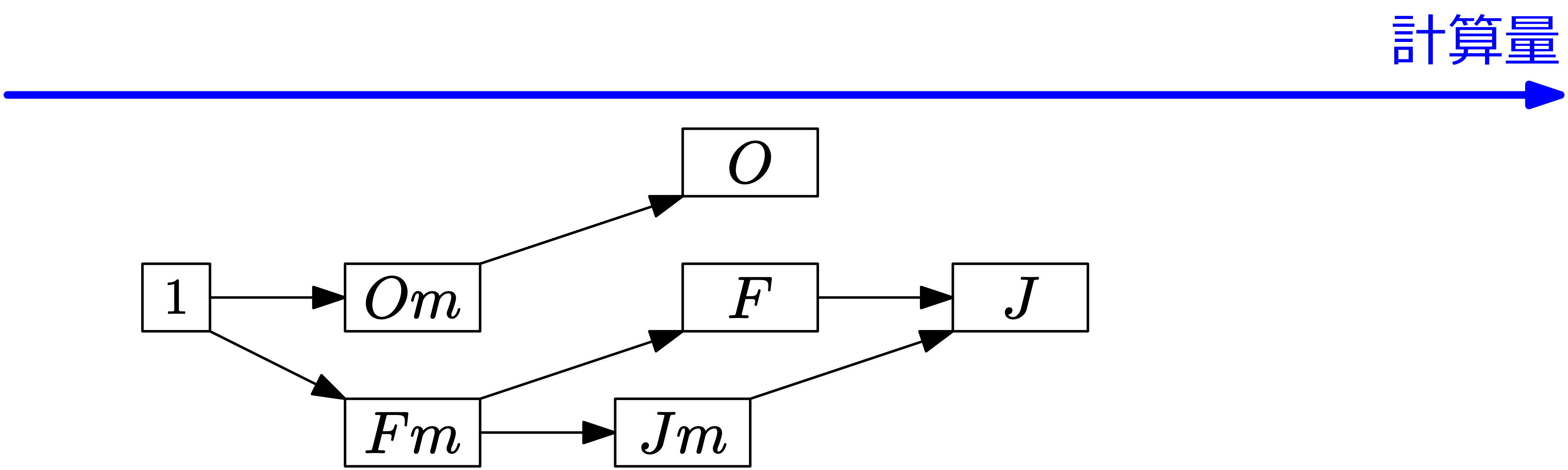
- O　オープンショップ　機械数は入力の一部
- F　フローショップ　機械数は入力の一部
- J　ジョブショップ　機械数は入力の一部

- Om　オープンショップ　機械数は m で固定
- Fm　フローショップ　機械数は m で固定
- Jm　ジョブショップ　機械数は m で固定



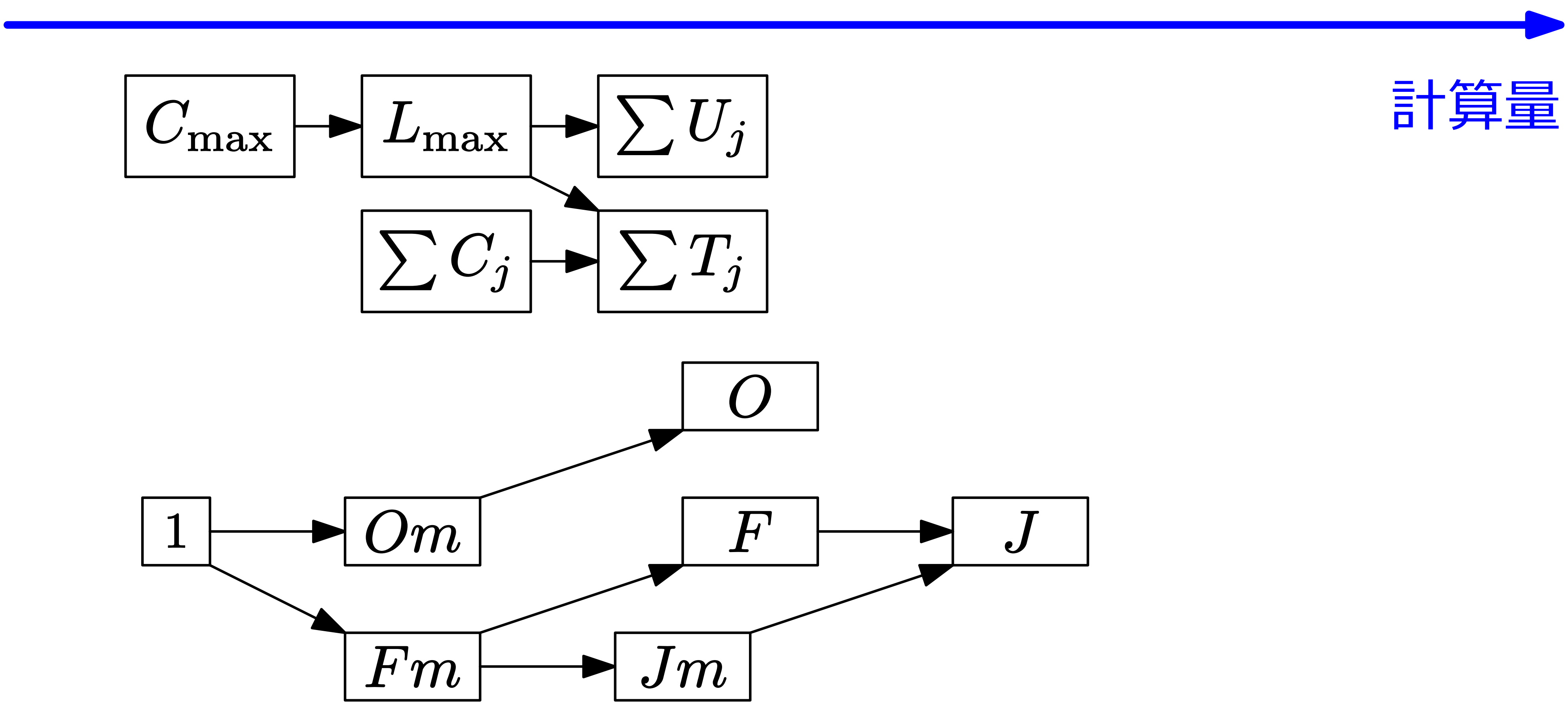
前回の内容

- $F2 \mid\mid C_{\max}$　$O(n \log n)$　(Johnson ’54)
- $F3 \mid\mid C_{\max}$　強 NP 困難　(Garey, Johnson, Sethi ’76)
- $J2 \mid\mid C_{\max}$　$O(n \log n)$　(Johnson ’54)



- $O2 \mid\mid C_{\max}$ $O(n)$ (Gonzalez, Sahni ’76)
- $O3 \mid\mid C_{\max}$ 弱 NP 困難 (Gonzalez, Sahni ’76)

- $O2 \mid\mid L_{\max}$ 強 NP 困難 (Lawler, Lenstra, Rinnooy Kan ’81)
- $F2 \mid\mid L_{\max}$ 強 NP 困難 (Lenstra, Rinnooy Kan, Brucker ’77)



本日の内容

1. **$O2 \mid\mid C_{\max}$ のアルゴリズム**
2. $O3 \mid\mid C_{\max}$ の弱 NP 困難性
3. $O2 \mid\mid L_{\max}$ の強 NP 困難性
4. $F2 \mid\mid L_{\max}$ の強 NP 困難性

- T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time. *Journal of the Association for Computing Machinery* 23 (1976) 665–679.

機械の環境

- オープンショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

$$
\begin{array}{c|ccccccc|}
 & J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} & O_{15} & O_{16} & O_{17} \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} & O_{25} & O_{26} & O_{27} \\
\hline
\end{array}
$$

機械の環境

- オープンショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

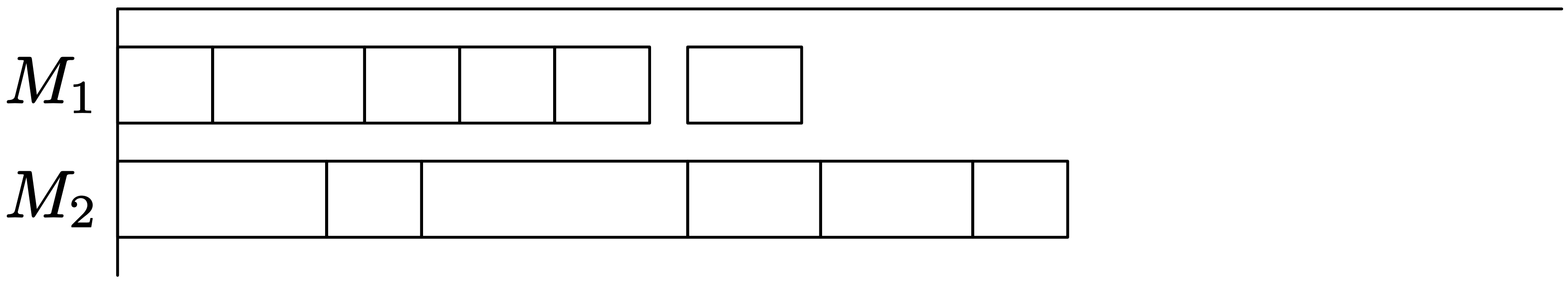
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2	5	1	3	2	0
M_2	1	5	3	2	3	5	6

$O2 \mid\mid C_{\max}$：最適値の下界

性質

$O2 \mid\mid C_{\max}$ において，最適値 $C^*_{\max}$ は次をすべて満たす

- $C^*_{\max} \geq \sum_{j=1}^{n} p_{1j}$
- $C^*_{\max} \geq \sum_{j=1}^{n} p_{2j}$
- $C^*_{\max} \geq \max\{p_{1j} + p_{2j} \mid j \in [n]\}$

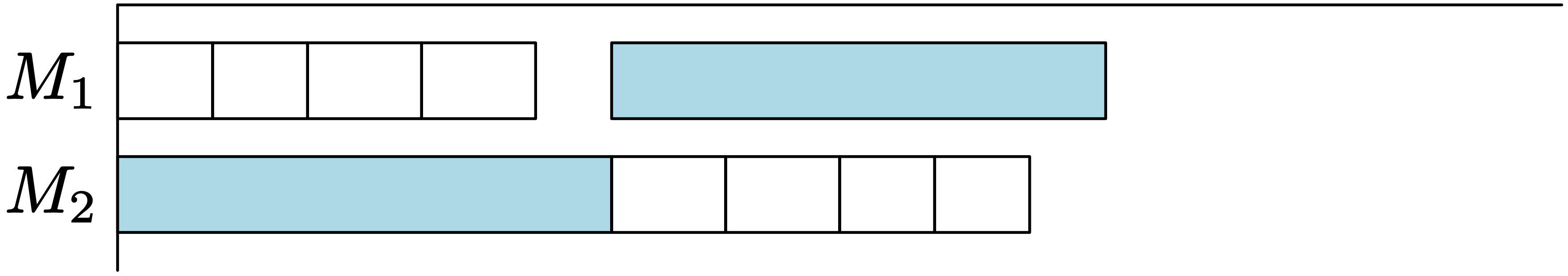


$O2 \mid\mid C_{\max}$：最適値の下界

性質

$O2 \mid\mid C_{\max}$ において，最適値 $C^*_{\max}$ は次をすべて満たす

- $C^*_{\max} \geq \displaystyle\sum_{j=1}^{n} p_{1j}$
- $C^*_{\max} \geq \displaystyle\sum_{j=1}^{n} p_{2j}$
- $C^*_{\max} \geq \max\{p_{1j} + p_{2j} \mid j \in [n]\}$

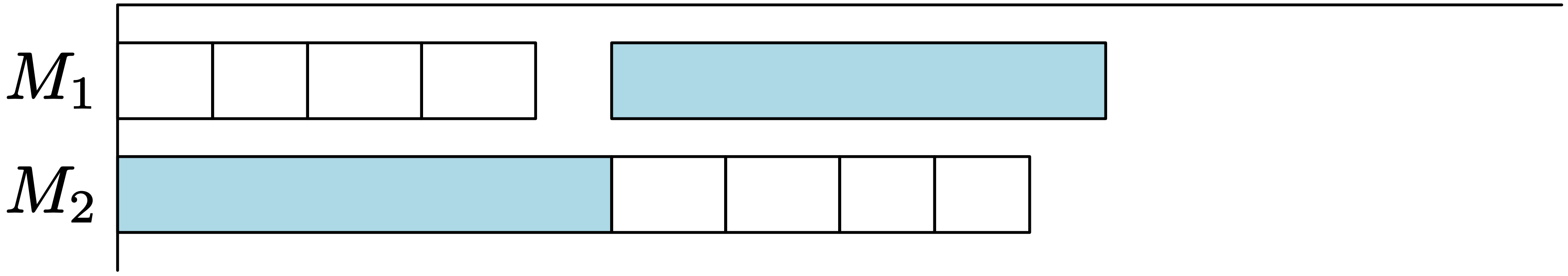


$O2 \mid\mid C_{\max}$ ：最適値の下界

性質

$O2 \mid\mid C_{\max}$ において，最適値 $C^*_{\max}$ は次をすべて満たす

- $C^*_{\max} \geq \sum_{j=1}^{n} p_{1j}$
- $C^*_{\max} \geq \sum_{j=1}^{n} p_{2j}$
- $C^*_{\max} \geq \max\{p_{1j} + p_{2j} \mid j \in [n]\}$



アイディア：$C^*_{\max}$ がこのどれかを等号で満たすようにする

$O2 \mid\mid C_{\max}$：アルゴリズムの動作例

	$\mathcal{J}_1$					$\mathcal{J}_2$				
M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2

M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2

$\mathcal{J}_1$ $\qquad\qquad$ $\mathcal{J}_2$

$O2 \mid\mid C_{\max}$：アルゴリズムの動作例

M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
	$\mathcal{J}_1$					$\mathcal{J}_2$				

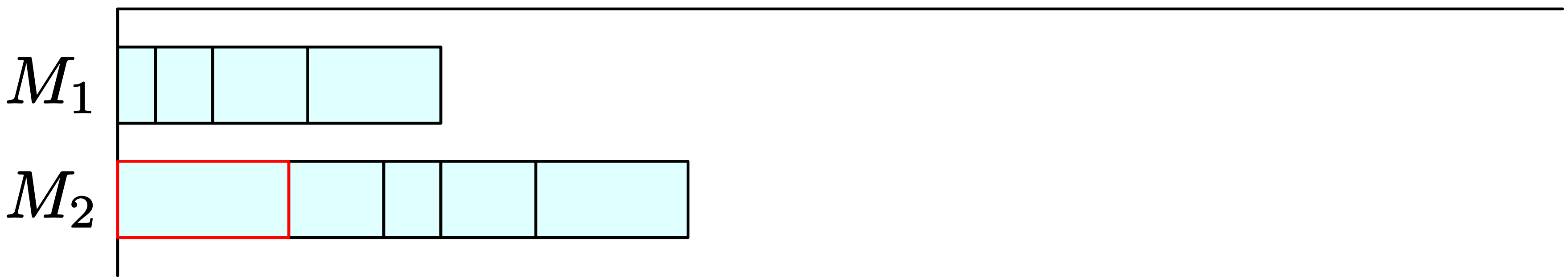
$O2 \mid\mid C_{\max}$：アルゴリズムの動作例

M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
			$\mathcal{J}_1$					$\mathcal{J}_2$		

$O2 \parallel C_{\max}$ ：アルゴリズムの動作例

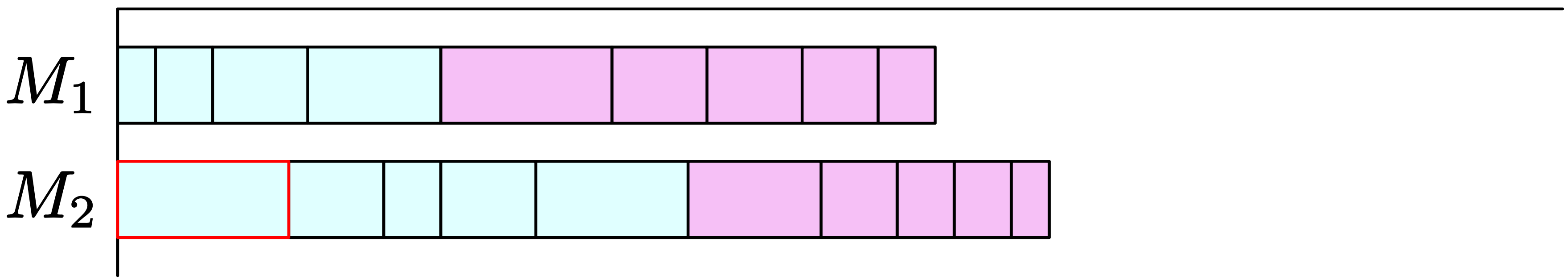
M_1	2	3	5	7	8	9	5	5	4	3
	$\le$	$\le$	$\le$	$\le$	$\le$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
	$\mathcal{J}_1$					$\mathcal{J}_2$				

$O2 \mid\mid C_{\max}$：アルゴリズムの動作例



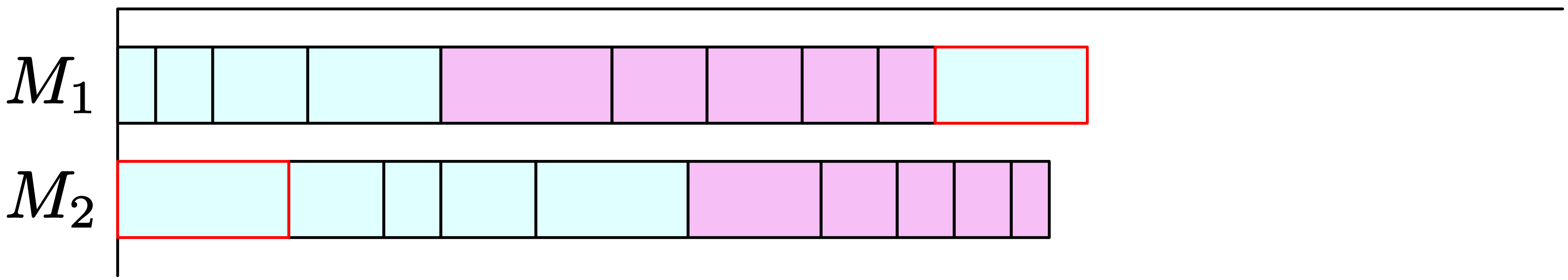
M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
			$\mathcal{J}_1$					$\mathcal{J}_2$		

$O2 \,||\, C_{\max}$：アルゴリズムの動作例



M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
	$\mathcal{J}_1$					$\mathcal{J}_2$				

$O2 \mid\mid C_{\max}$：アルゴリズムの動作例



	$\mathcal{J}_1$					$\mathcal{J}_2$				
M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2

$O2 \mid\mid C_{\max}$：アルゴリズム

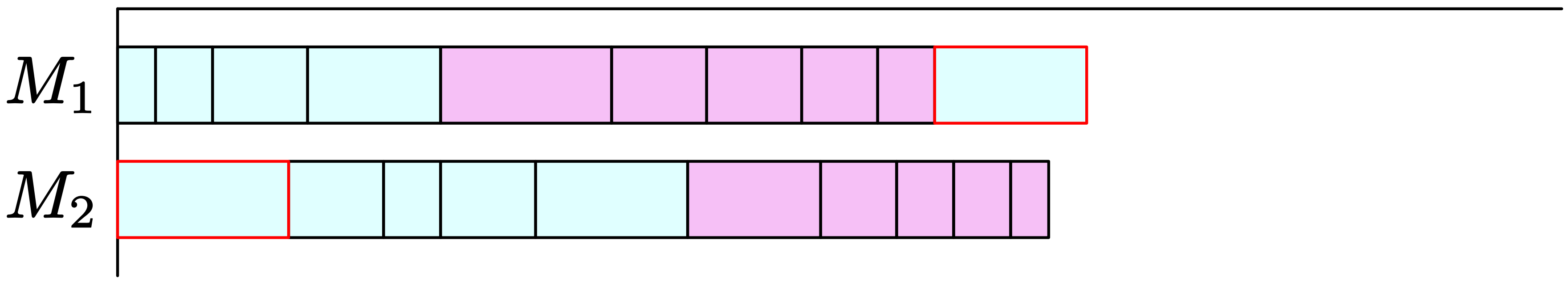
$O2 \mid\mid C_{\max}$ のアルゴリズム

1. $\mathcal{J}_1 = \{J_j \mid p_{1j} \le p_{2j}\}$, $\mathcal{J}_2 = \{J_j \mid p_{1j} > p_{2j}\}$ とする
2. $\max\{\max\{p_{1j} \mid J_j \in \mathcal{J}_1\}, \max\{p_{2j} \mid J_j \in \mathcal{J}_2\}\}$ を与えるジョブ J を見つける (以下，$J \in \mathcal{J}_1$ と仮定)
3. M_1 では，次の順でジョブを処理する
   - $\mathcal{J}_1 - \{J\}$ を任意順，$\mathcal{J}_2$ を任意順，J
4. M_2 では，次の順でジョブを処理する
   - J，$\mathcal{J}_1 - \{J\}$ を M_1 と同順，$\mathcal{J}_2$ を M_1 と同順

M_1	2	3	5	7	8	9	5	5	4	3
	$\le$	$\le$	$\le$	$\le$	$\le$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
	$\mathcal{J}_1$					$\mathcal{J}_2$				

$O2 \mid\mid C_{\max}$ のアルゴリズム

1. $\mathcal{J}_1 = \{J_j \mid p_{1j} \le p_{2j}\}$, $\mathcal{J}_2 = \{J_j \mid p_{1j} > p_{2j}\}$ とする
2. $\max\{\max\{p_{1j} \mid J_j \in \mathcal{J}_1\}, \max\{p_{2j} \mid J_j \in \mathcal{J}_2\}\}$ を与えるジョブ J を見つける (以下，$J \in \mathcal{J}_1$ と仮定)
3. M_1 では，次の順でジョブを処理する
   - $\mathcal{J}_1 - \{J\}$ を任意順，$\mathcal{J}_2$ を任意順，J
4. M_2 では，次の順でジョブを処理する
   - J，$\mathcal{J}_1 - \{J\}$ を M_1 と同順，$\mathcal{J}_2$ を M_1 と同順



定理 (Gonzalez, Sahni '76)

前のページのアルゴリズムで，$O2 \mid\mid C_{\max}$ は $O(n)$ 時間で解ける

n はジョブ数

定理 (Gonzalez, Sahni ’76)

前のページのアルゴリズムで，$O2 \,||\, C_{\max}$ は
$O(n)$ 時間で解ける n はジョブ数

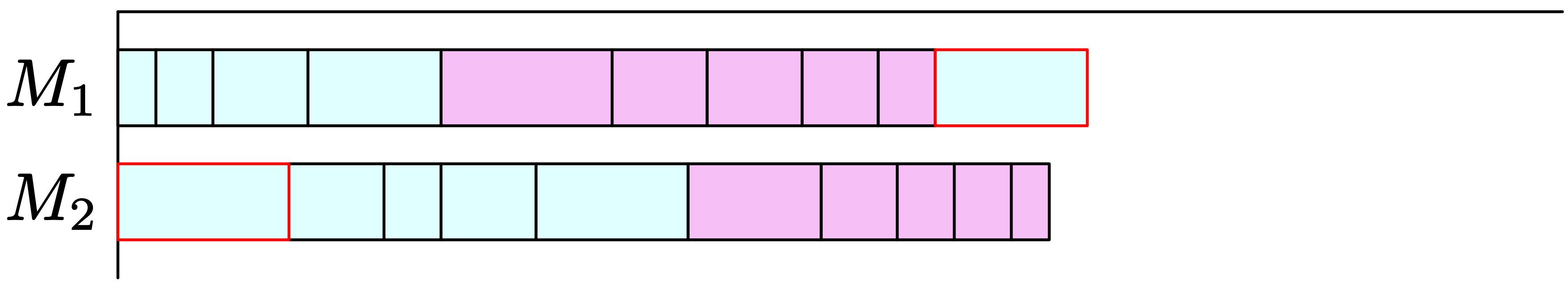
WLOG ジョブには次のように番号がついているとする
- $\mathcal{J}_1 - \{J_{n_1}\}$ は $J_1, J_2, \ldots, J_{n_1-1}$ の順に処理
- $\mathcal{J}_2$ は $J_{1'}, J_{2'}, \ldots, J_{n'_2}$ の順に処理

	1	2		n_1-1	n_1	$1'$	$2'$		$(n_2-1)'$	n'_2
M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2

$\mathcal{J}_1$ $\mathcal{J}_2$

M_1 の処理順：$1 \to \cdots \to n_1{-}1 \to 1' \to \cdots \to n_2' \to \boxed{n_1}$

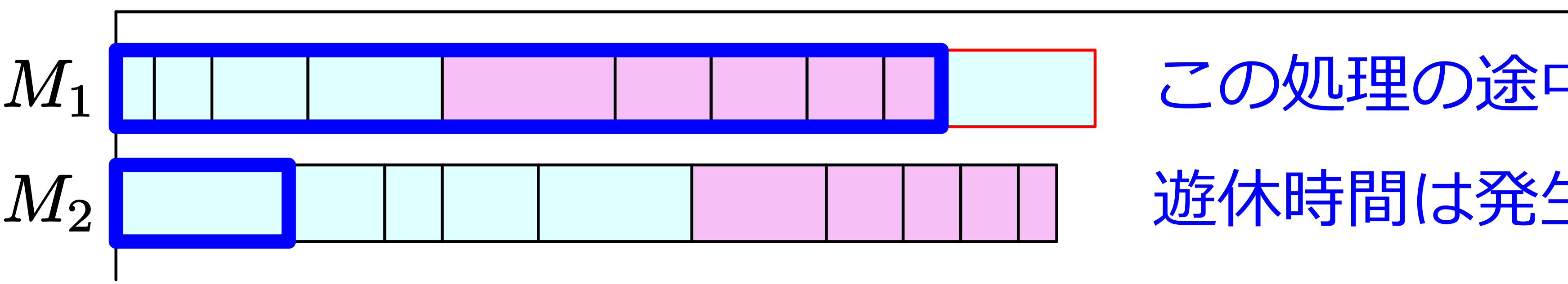
M_2 の処理順：$\boxed{n_1} \to 1 \to \cdots \to n_1{-}1 \to 1' \to \cdots \to n_2'$



	1	2		$n_1{-}1$	n_1	$1'$	$2'$		$(n_2{-}1)'$	n_2'
M_1	2	3	5	7	8	9	5	5	4	3
	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
			$\mathcal{J}_1$					$\mathcal{J}_2$		

M_1 の処理順：$1 \to \cdots \to n_1{-}1 \to 1' \to \cdots \to n_2' \to \boxed{n_1}$

M_2 の処理順：$\boxed{n_1} \to 1 \to \cdots \to n_1{-}1 \to 1' \to \cdots \to n_2'$



この処理の途中で

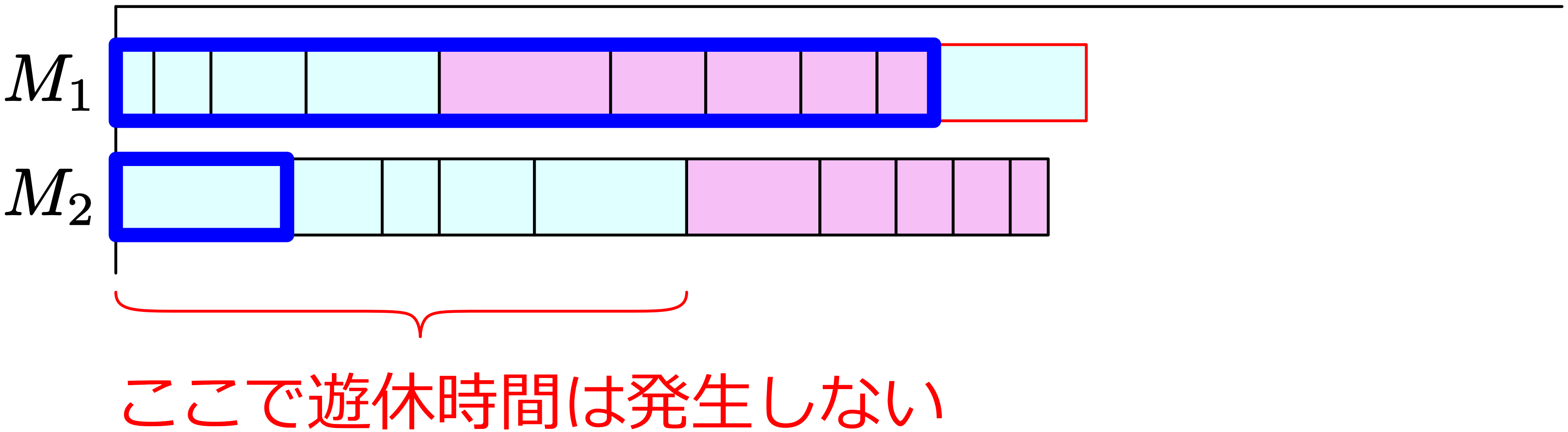
遊休時間は発生しない

	1	2		$n_1{-}1$	n_1	$1'$	$2'$		$(n_2{-}1)'$	n_2'
M_1	2	3	5	7	8	9	5	5	4	3
	$\le$	$\le$	$\le$	$\le$	$\le$	$>$	$>$	$>$	$>$	$>$
M_2	5	3	5	8	9	7	4	3	3	2
			$\mathcal{J}_1$					$\mathcal{J}_2$		

補題

M_2 で $\mathcal{J}_1$ のジョブを処理している途中で，
遊休時間は発生しない

証明：

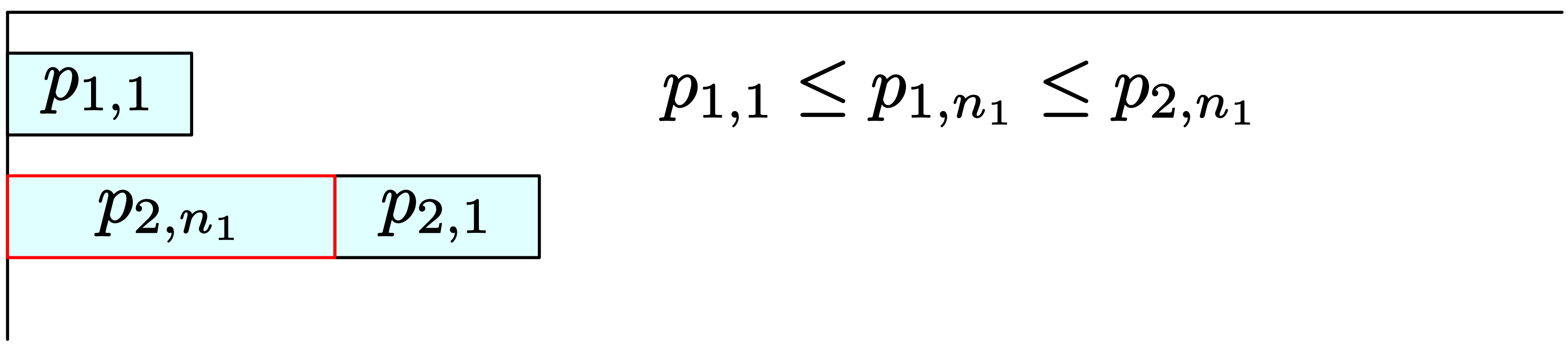


$O2 \mid\mid C_{\max}$: 証明 (2)

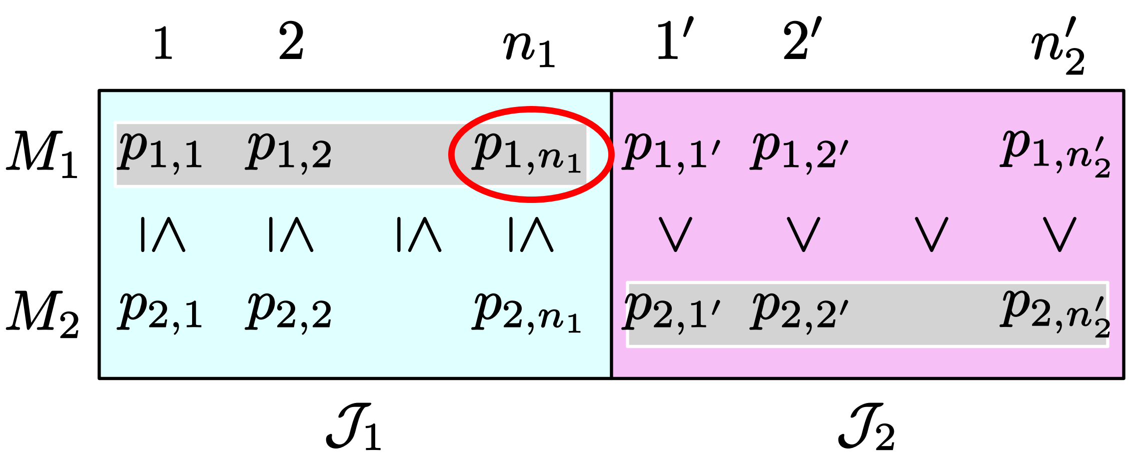
補題

M_2 で $\mathcal{J}_1$ のジョブを処理している途中で，
遊休時間は発生しない

証明：



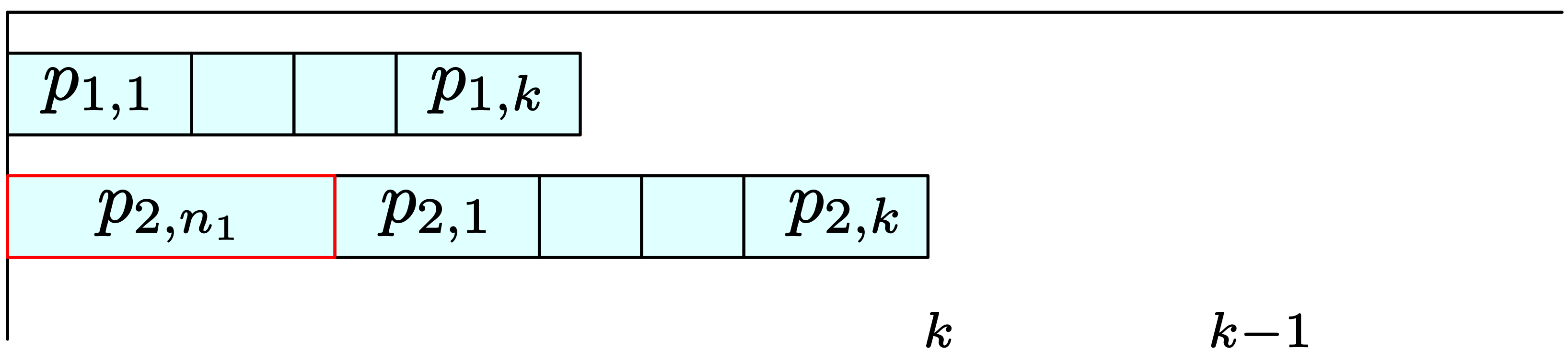
$$p_{1,1} \leq p_{1,n_1} \leq p_{2,n_1}$$

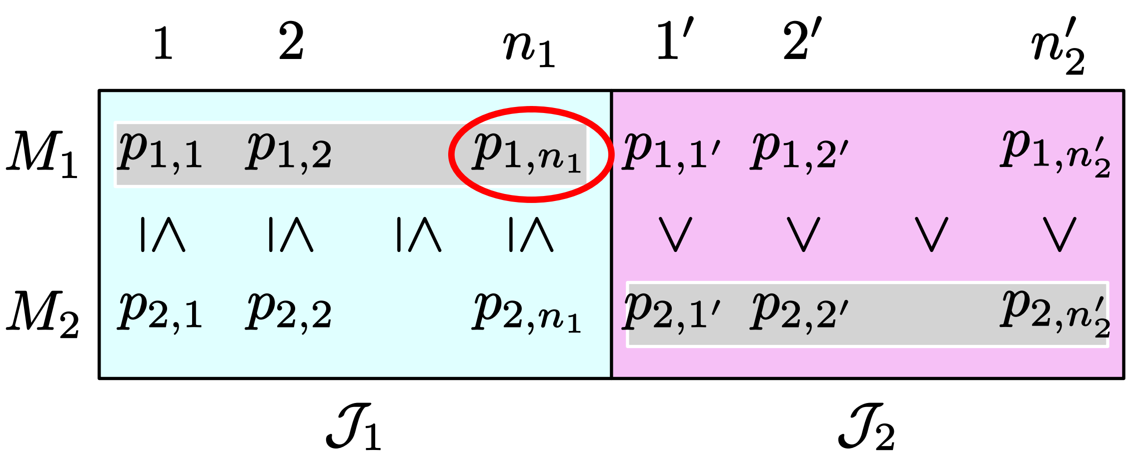


補題

M_2 で $\mathcal{J}_1$ のジョブを処理している途中で，
遊休時間は発生しない

<u>証明</u>：

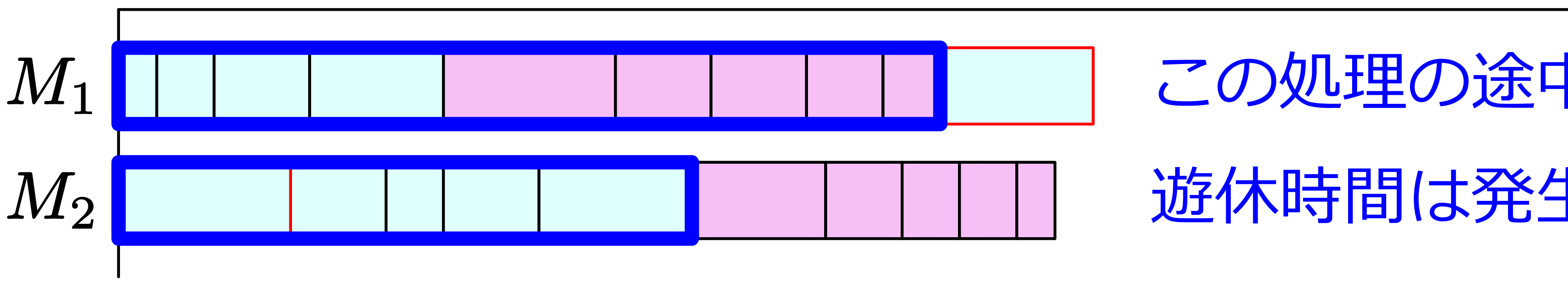




$$\sum_{j=1}^{k} p_{1,j} = \sum_{j=1}^{k-1} p_{1,j} + p_{1,k} \leq \sum_{j=1}^{k-1} p_{2,j} + p_{1,n_1}$$

□

$O2 \,||\, C_{\max}$：証明 (3)



この処理の途中で
遊休時間は発生しない

3 つに場合分け

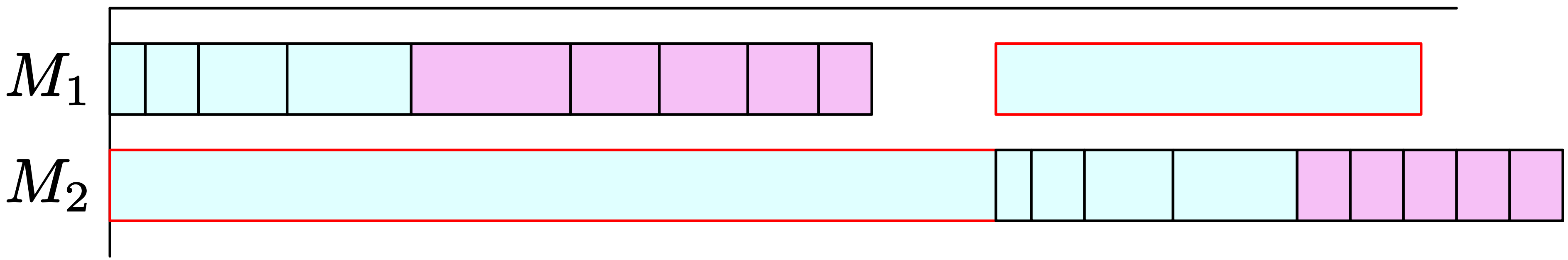
(A) 処理の途中で，どこにも遊休時間がない場合

- アルゴリズムの $C_{\max} = \max\left\{\sum p_{1j}, \sum p_{2j}\right\}$
- したがって，アルゴリズムの出力は最適解

復習：$C^*_{\max} \geq \sum p_{1j}, C^*_{\max} \geq \sum p_{2j}, C^*_{\max} \geq p_{1j} + p_{2j}$

$O2 \mid\mid C_{\max}$：証明 (4)

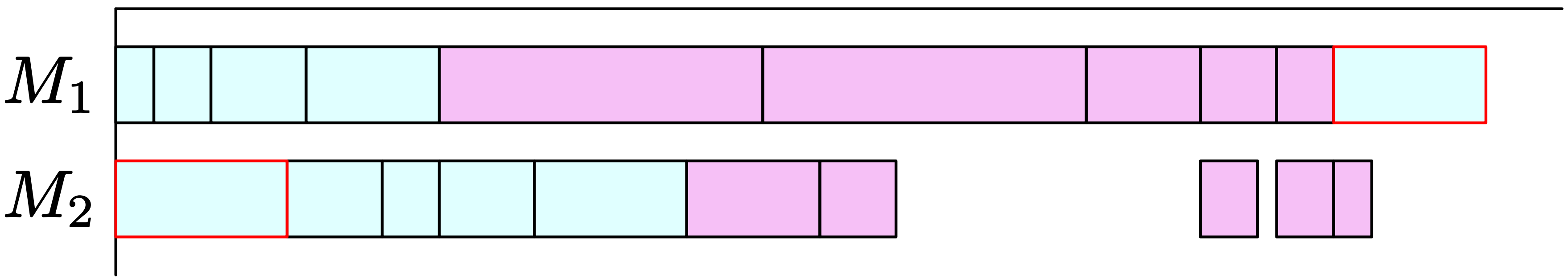
(B) M_1 で J_{n_1} を処理するときに遊休時間が必要である場合



- アルゴリズムの $C_{\max} = \max\left\{p_{1,n_1} + p_{2,n_1}, \sum p_{2j}\right\}$
- したがって，アルゴリズムの出力は最適解

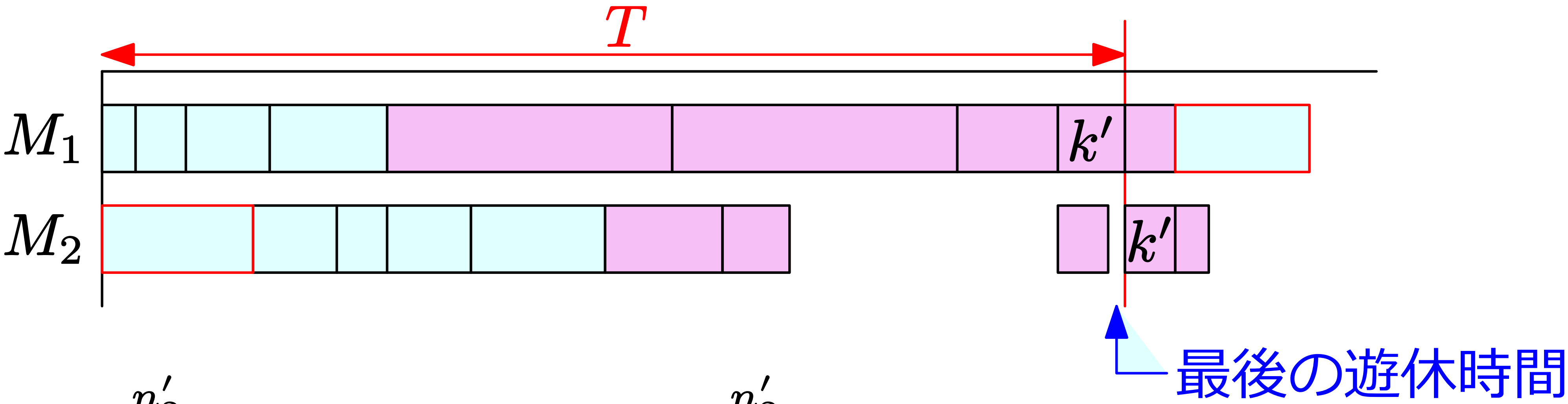
復習：$C^*_{\max} \geq \sum p_{1j}, C^*_{\max} \geq \sum p_{2j}, C^*_{\max} \geq p_{1j} + p_{2j}$

(C) M_2 で $\mathcal{J}_2$ のジョブを処理するときに遊休時間がある場合



$O2 \mid\mid C_{\max}$: 証明 (5)

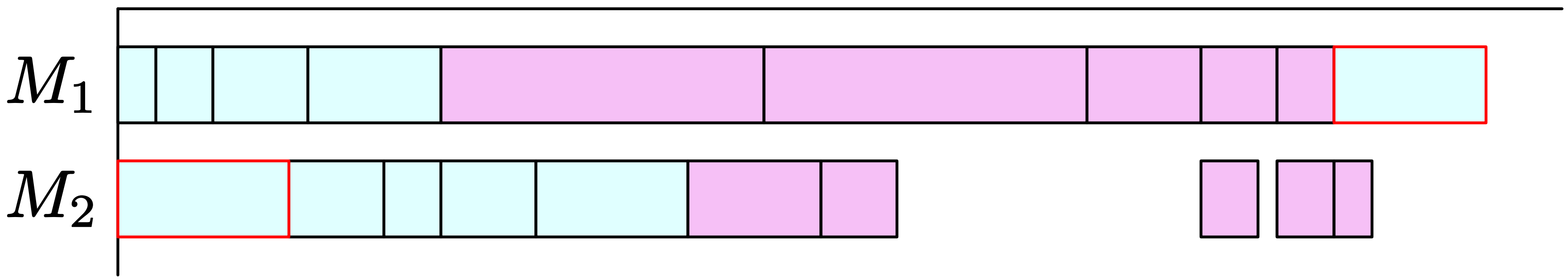
(C) M_2 で $\mathcal{J}_2$ のジョブを処理するときに遊休時間がある場合



$$\sum_{j'=(k+1)'}^{n'_2} p_{1,j'} + p_{1,n_1} > \sum_{j'=(k+1)'}^{n'_2} p_{2,j'} + p_{2,k'}$$

	1	2		n_1	$1'$	$2'$		n'_2
M_1	$p_{1,1}$	$p_{1,2}$		p_{1,n_1}	$p_{1,1'}$	$p_{1,2'}$		p_{1,n'_2}
	$\leq$	$\leq$	$\leq$	$\leq$	$>$	$>$	$>$	$>$
M_2	$p_{2,1}$	$p_{2,2}$		p_{2,n_1}	$p_{2,1'}$	$p_{2,2'}$		p_{2,n'_2}
	$\mathcal{J}_1$				$\mathcal{J}_2$			

(C) M_2 で $\mathcal{J}_2$ のジョブを処理するときに遊休時間がある場合



- アルゴリズムの $C_{\max} = \sum p_{1j}$
- したがって，アルゴリズムの出力は最適解

$\therefore$ 3 つのどの場合でも，アルゴリズムの出力は最適解 $\square$

復習：$C^*_{\max} \geq \sum p_{1j}, C^*_{\max} \geq \sum p_{2j}, C^*_{\max} \geq p_{1j} + p_{2j}$

本日の内容

1. $O2 \,||\, C_{\max}$ のアルゴリズム
2. **$O3 \,||\, C_{\max}$ の弱 NP 困難性**
3. $O2 \,||\, L_{\max}$ の強 NP 困難性
4. $F2 \,||\, L_{\max}$ の強 NP 困難性

- T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time. *Journal of the Association for Computing Machinery* 23 (1976) 665–679.

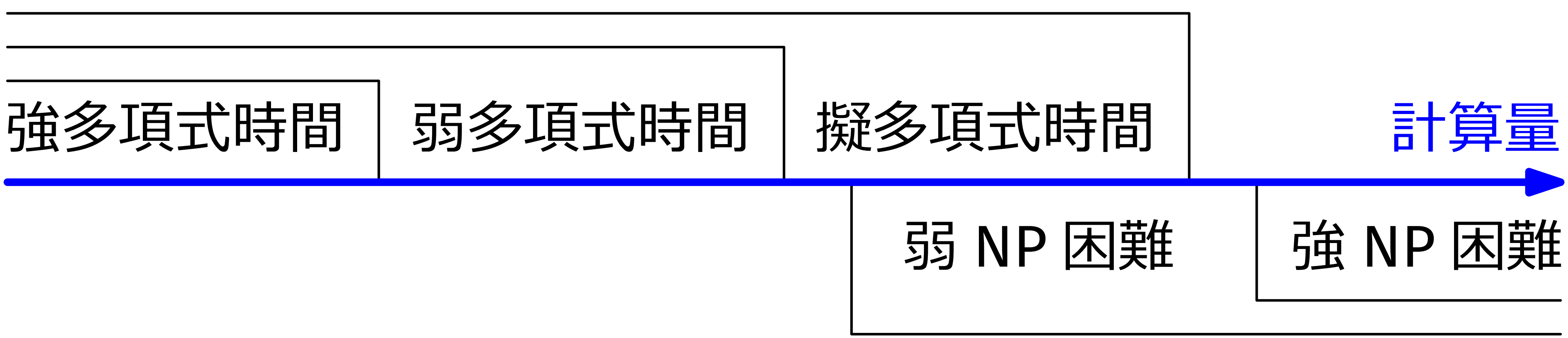
$O3 \mid\mid C_{\max}$ の弱 NP 困難性

定理 (Gonzalez, Sahni ’76)

問題 $O3 \mid\mid C_{\max}$ は 弱 NP 困難

注 1：$O2 \mid\mid C_{\max}$ は強多項式時間で解ける

注 2：$O3 \mid\mid C_{\max}$ が強 NP 困難か 未解決



[復習] 分割問題

定義：分割問題 (partition problem)

正整数 $a_1, a_2, \ldots, a_n$ が与えられたとき,
それらを同じ和の 2 つのグループに分けられるか？

例：$\underline{1, 1, 2, 2,}\ \underline{3, 4,}\ \underline{6,}\ \underline{6, 7,}\ \underline{8}$

- $1 + 1 + 2 + 2 + 6 + 8 \quad = 20$
- $3 + 4 + 6 + 7 \quad = 20$

できる

例：$1, 1, 2, 3, 4, 5, 6, 6, 7, 10$ 　できない

事実 (Karp ’72)

分割問題は 弱 NP 困難

証明：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

$$T = \frac{1}{2}\sum_{i=1}^{n} a_i$$

M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

分割問題を解くアルゴリズム

$O3 \,||\, C_{\max}$：証明 (1)

証明：分割問題を帰着する

入力の変換：$1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

$$T = \frac{1}{2}\sum_{i=1}^{n} a_i$$

M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

a_i

0

0

分割問題を解くアルゴリズム

<u>証明</u>：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

$$T = \frac{1}{2}\sum_{i=1}^{n} a_i$$

M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

0

a_i

0

分割問題を解くアルゴリズム

証明：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

$$T = \frac{1}{2}\sum_{i=1}^{n} a_i$$

M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

0

0

a_i

分割問題を解くアルゴリズム

証明：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

$$T = \frac{1}{2}\sum_{i=1}^{n} a_i$$

M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

T
T
T

分割問題を解くアルゴリズム

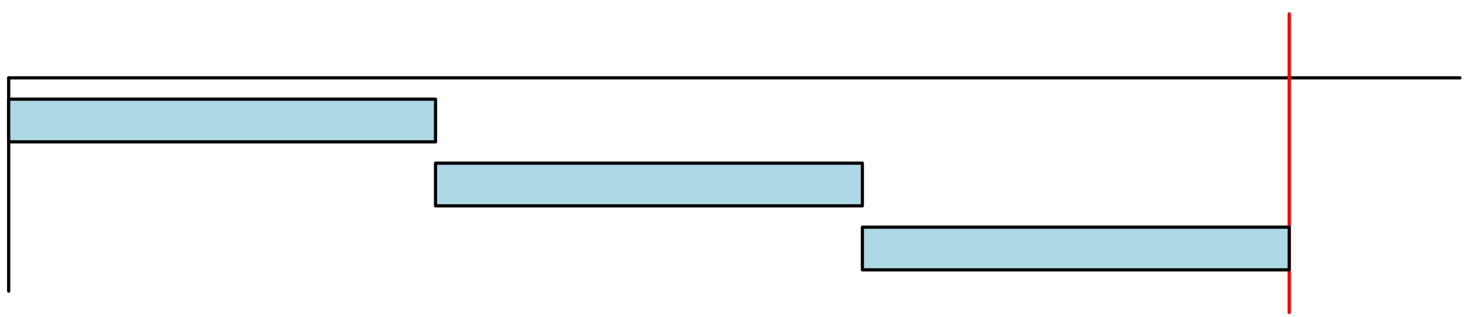
$O3 \mid\mid C_{\max}$：証明 (2)

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



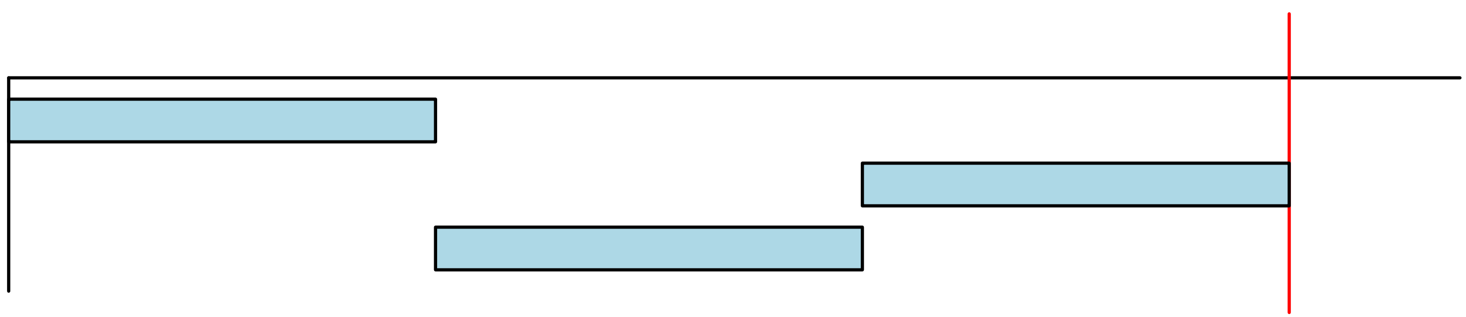
M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 □



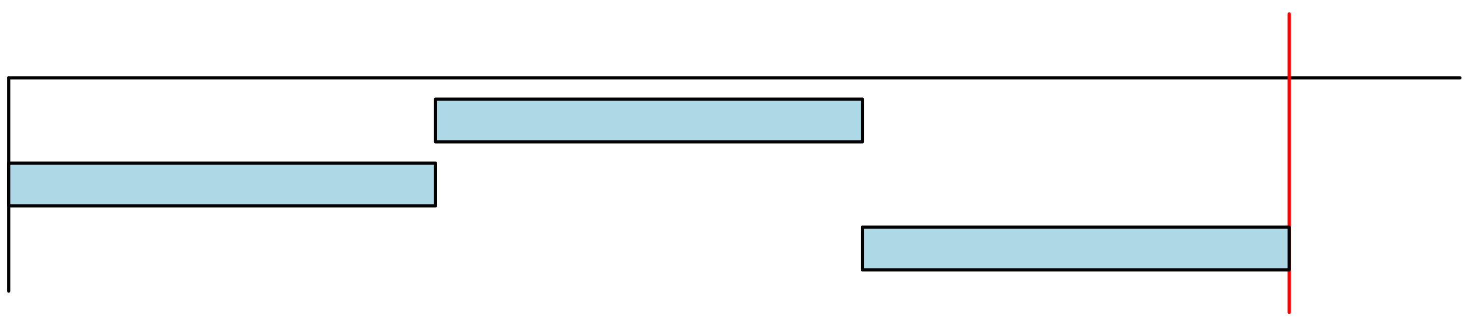
M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

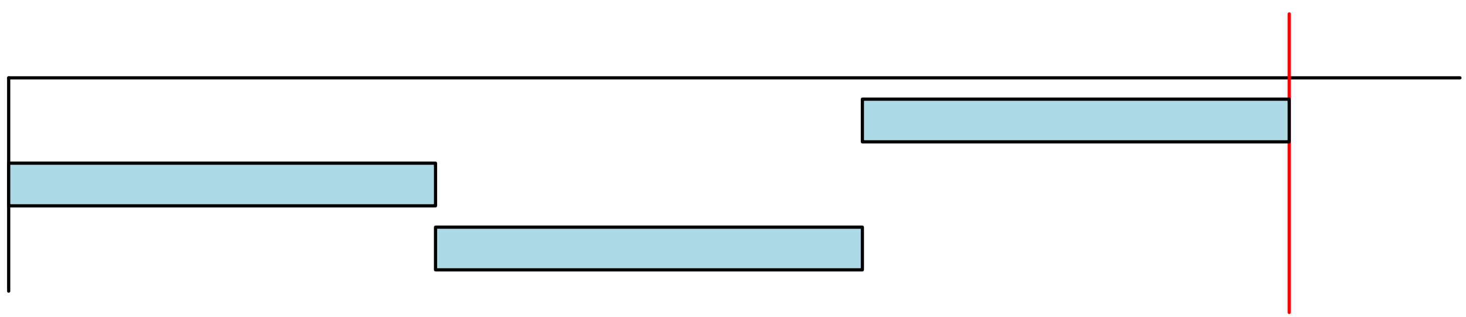
$O3 \mid\mid C_{\max}$：証明 (2)

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

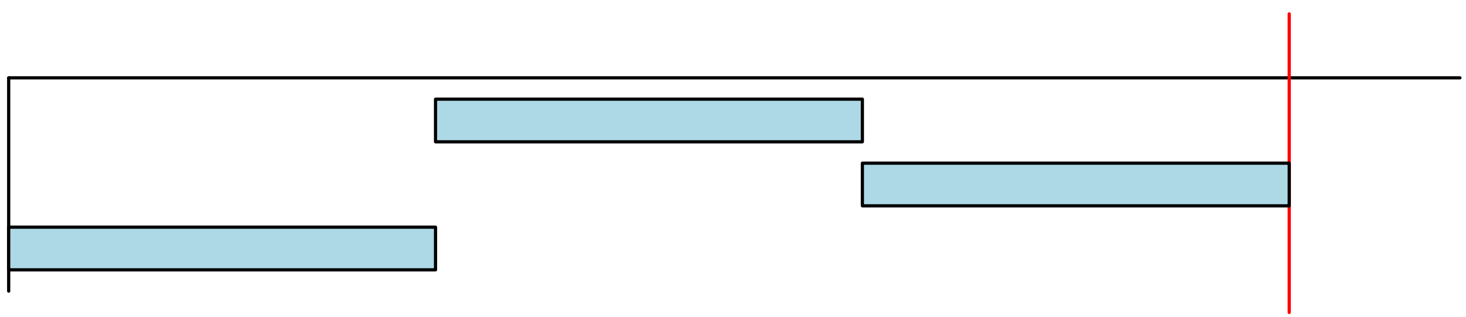
$O3 \mid\mid C_{\max}$：証明 (2)

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



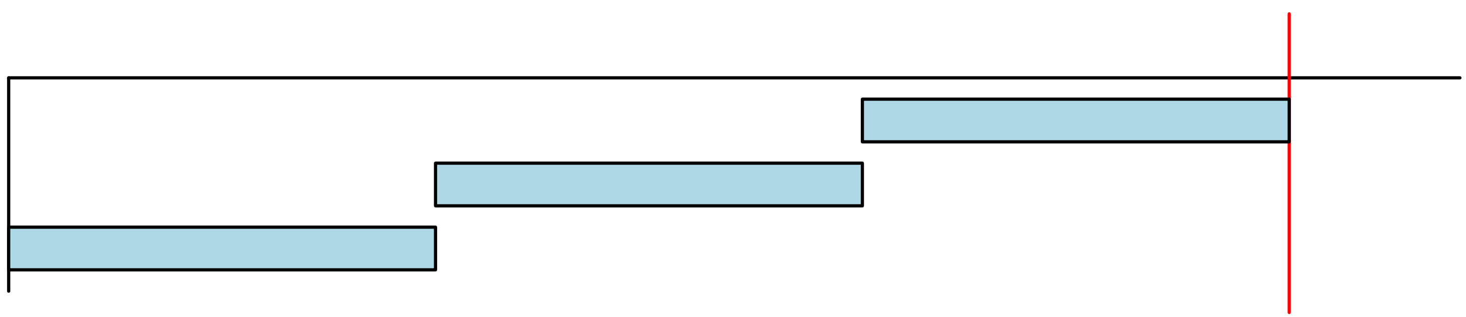
M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



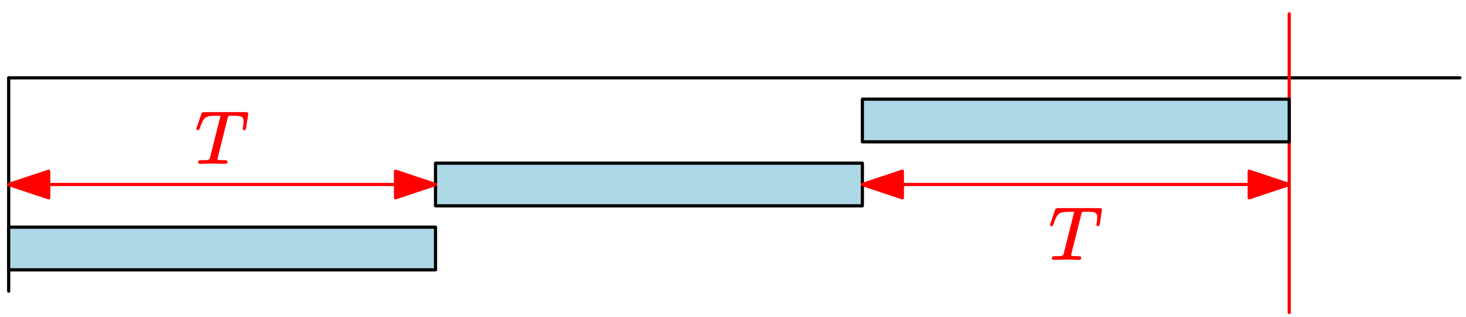
M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

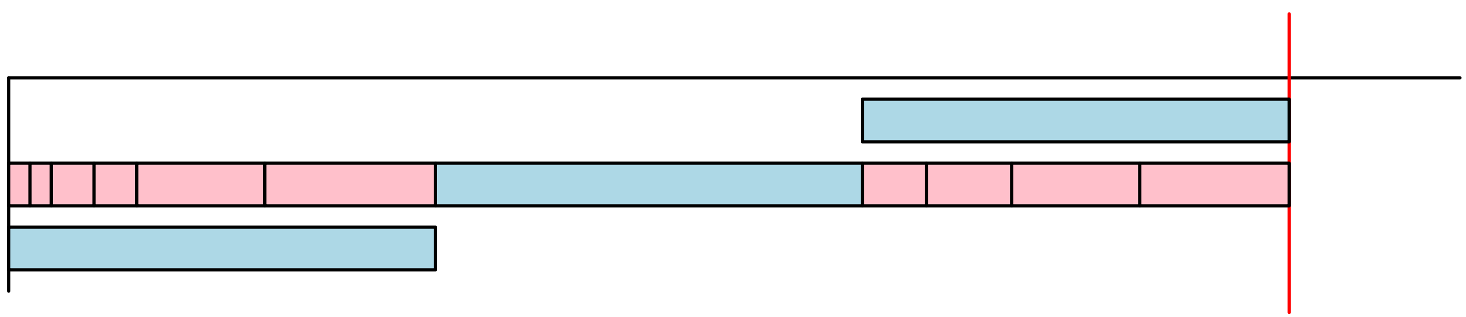
$O3 \mid\mid C_{\max}$: 証明 (2)

証明：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

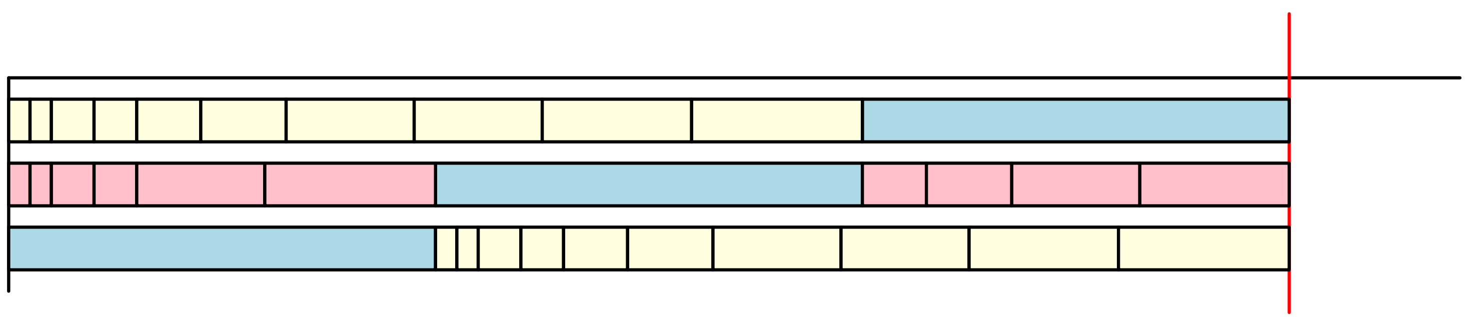
$O3 \mid\mid C_{\max}$：証明 (2)

<u>証明</u>：分割問題を帰着する

出力の変換：

最適値 $\leq 3T \Rightarrow$ 分割問題の答えは「できる」

最適値 $> 3T \Rightarrow$ 分割問題の答えは「できない」 $\square$



M_1	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
M_2	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	0	0	0	0	0	0	0	0	0	0	20
M_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	6	6	7	8	20

本日の内容

1. $O2 \mid\mid C_{\max}$ のアルゴリズム
2. $O3 \mid\mid C_{\max}$ の弱 NP 困難性
3. $O2 \mid\mid L_{\max}$ **の強 NP 困難性**
4. $F2 \mid\mid L_{\max}$ の強 NP 困難性

- E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Minimizing maximum lateness in a two-machine open shop. *Mathematics of Operations Research* 6 (1981) 153–158. Erratum: 7 (1982) 635.

機械の環境

- オープンショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間
- $d_j =$ ジョブ J_j の納期

最適化の目的

- 最大納期ずれの最小化

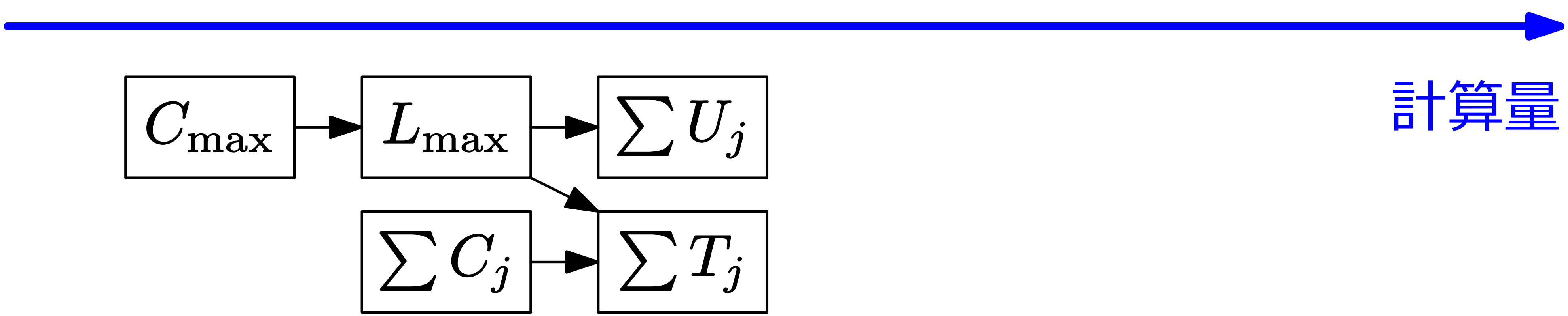
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2	5	1	3	2	0
M_2	1	5	3	2	3	5	6
d_j	8	10	10	7	9	16	12

定理 (Lawler, Lenstra, Rinnooy Kan ’81)

問題 $O2 \mid\mid L_{\max}$ は強 NP 困難

注：$O2 \mid\mid C_{\max}$ は強多項式時間で解ける

$1 \mid\mid L_{\max}$ は強多項式時間で解ける (EDD 順)



[復習] 3 分割問題

定義：3 分割問題 (3-partition problem)

次を満たす正整数 $a_1, a_2, \ldots, a_{3n}$ が与えられる

$$\frac{1}{4}T < a_i < \frac{1}{2}T \quad \text{ただし，} \; T = \frac{1}{n}\sum_{i=1}^{3n} a_i$$

それらを同じ和の n 個のグループに分けられるか？

例：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

- $120 + 125 + 145 = 390$
- $123 + 127 + 140 = 390$
- $119 + 122 + 149 = 390$
- $130 + 130 + 130 = 390$

できる

事実 (Garey, Johnson ’75)

3 分割問題 は強 NP 困難

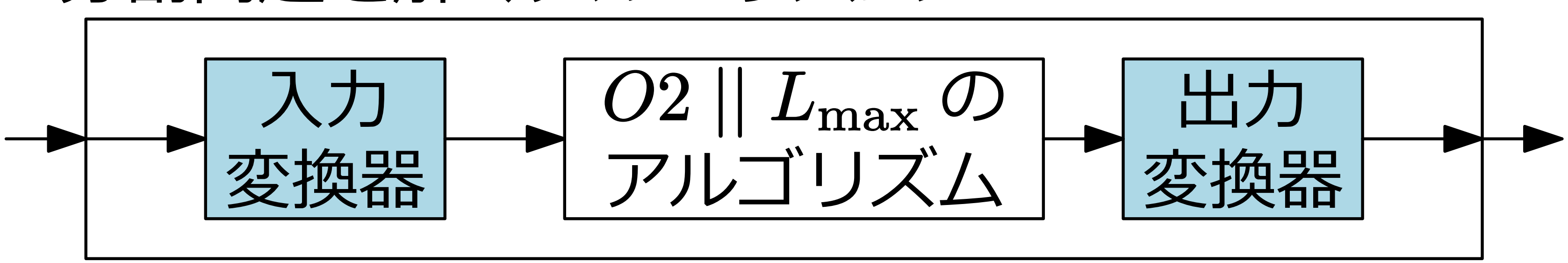
証明：3 分割問題を帰着する

$n = 4, T = 390$

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

3 分割問題を解くアルゴリズム



証明：3 分割問題を帰着する

$n = 4, T = 390$

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

↓

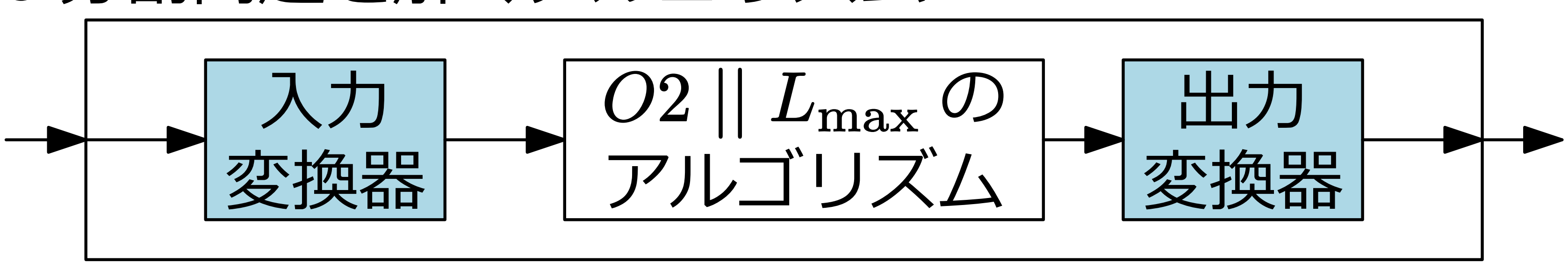
M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

0

a_i

$n(T+1)$

3 分割問題を解くアルゴリズム



$O2 \mid\mid L_{\max}$：証明 (1)

証明：3 分割問題を帰着する

$n = 4, T = 390$

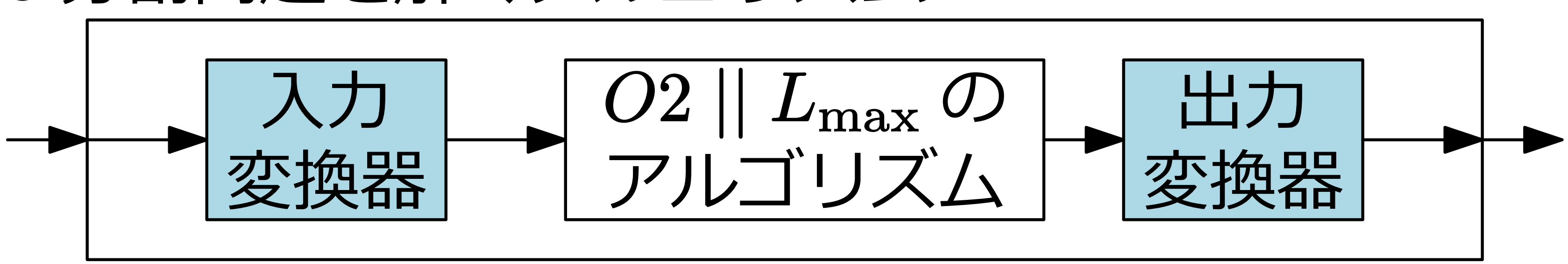
入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

↓

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

0
1
1

3 分割問題を解くアルゴリズム



$O2 \mid\mid L_{\max}$：証明 (1)

証明：3 分割問題を帰着する

$n = 4, T = 390$

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

$n-1$ 個

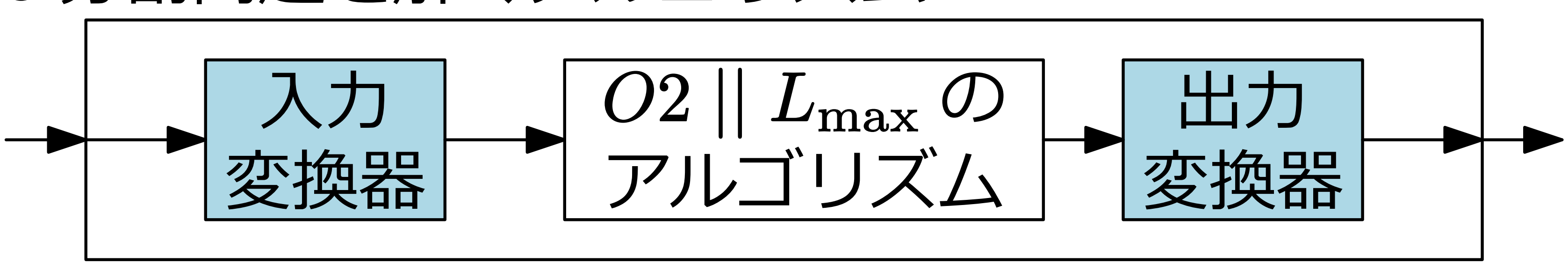
M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

$T+1$

1

$iT+i+1$

3 分割問題を解くアルゴリズム



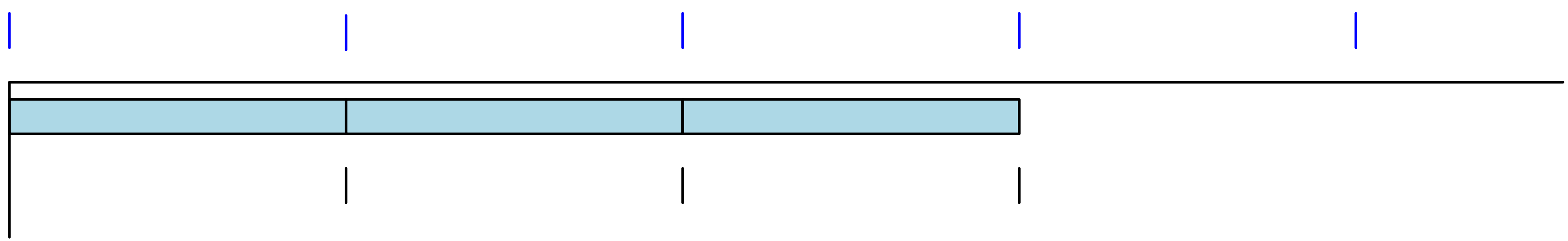
証明：3 分割問題を帰着する

出力の変換：

最適値 ≤ 0 $\Rightarrow$ 3 分割問題の答えは「できる」

最適値 > 0 $\Rightarrow$ 3 分割問題の答えは「できない」

□



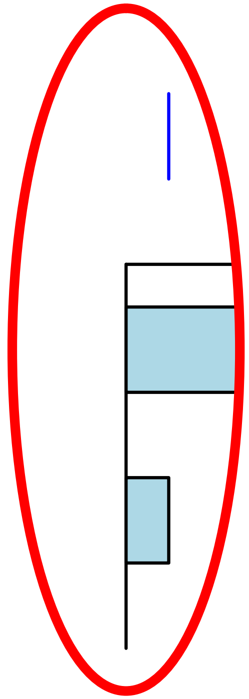
M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

証明：3 分割問題

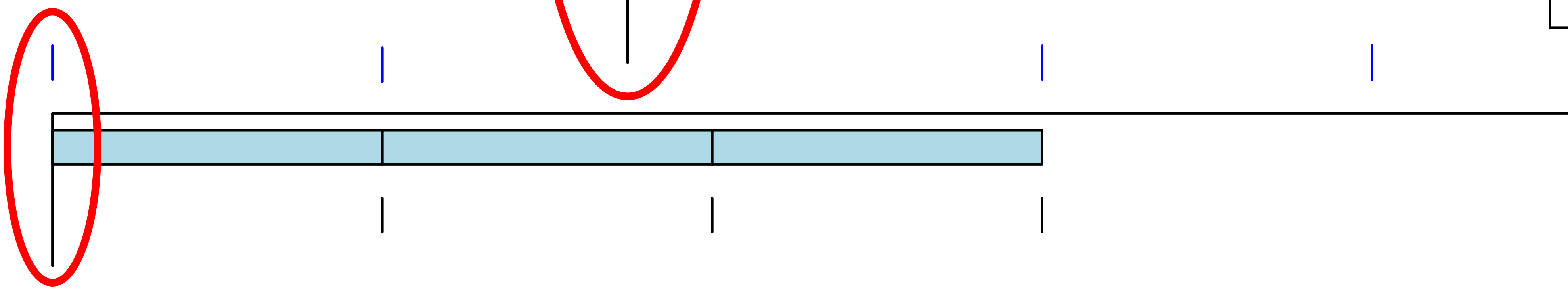
出力の変換：

最適値 ≤ 0 答えは「できる」

最適値 > 0 答えは「できない」



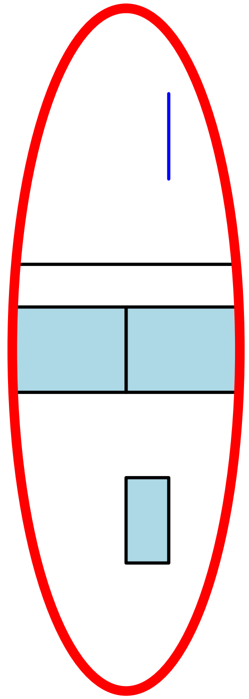
□



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

証明：3 分割問題ｱ

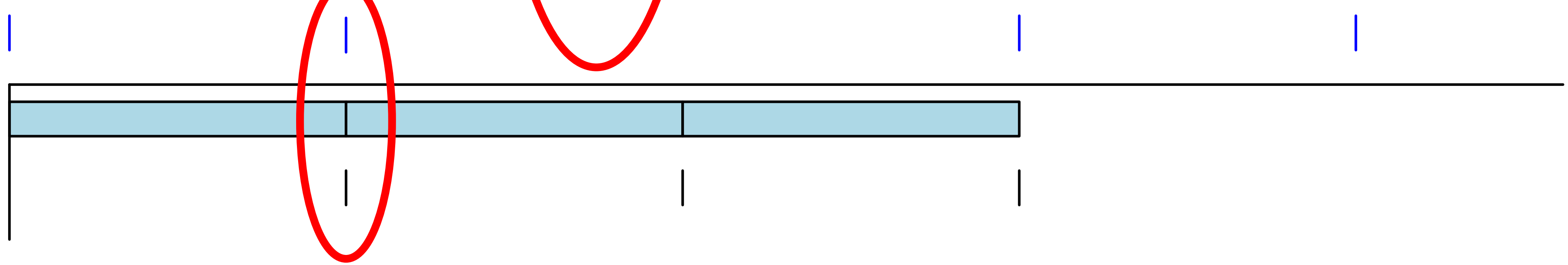
出力の変換：



最適値 ≤ 0 答えは「できる」

最適値 > 0 答えは「できない」

□



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

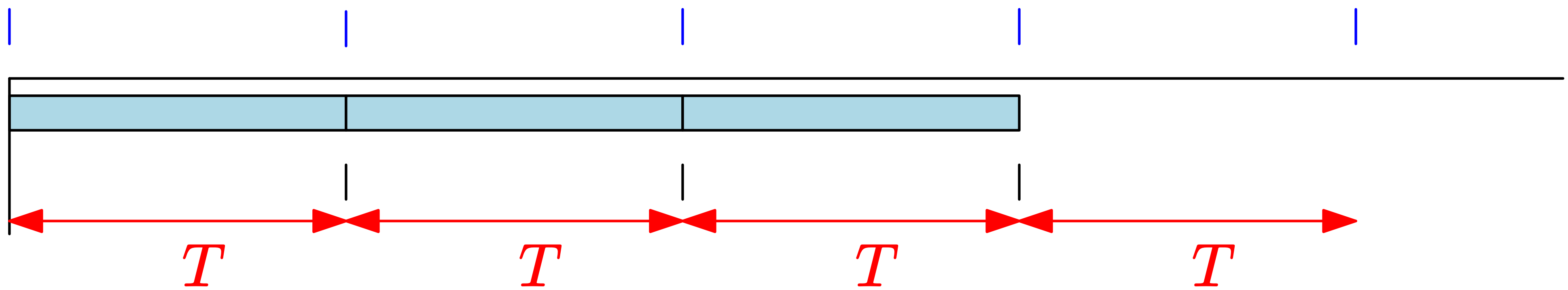
証明：3 分割問題を帰着する

出力の変換：

最適値 ≤ 0 $\Rightarrow$ 3 分割問題の答えは「できる」

最適値 > 0 $\Rightarrow$ 3 分割問題の答えは「できない」

$\square$



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

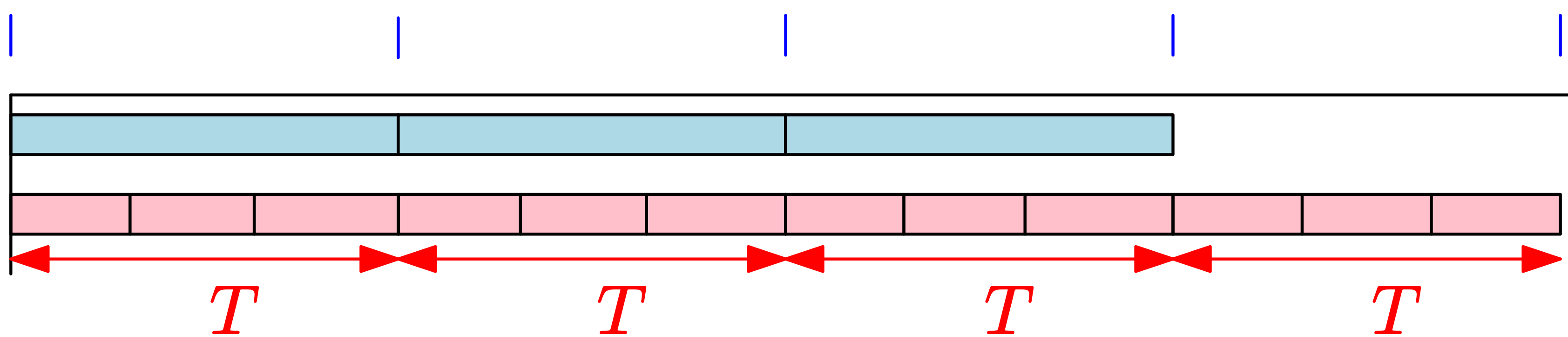
<u>証明</u>：3 分割問題を帰着する

出力の変換：

最適値 ≤ 0 $\Rightarrow$ 3 分割問題の答えは「できる」

最適値 > 0 $\Rightarrow$ 3 分割問題の答えは「できない」

□



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

本日の内容

1. $O2 \mid\mid C_{\max}$ のアルゴリズム
2. $O3 \mid\mid C_{\max}$ の弱 NP 困難性
3. $O2 \mid\mid L_{\max}$ の強 NP 困難性
4. **$F2 \mid\mid L_{\max}$ の強 NP 困難性**

- J.K. Lenstra, A.H.G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems. *Annals of Discrete Mathematics* 1 (1977) 343–362.

機械の環境

- フローショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間
- $d_j =$ ジョブ J_j の納期

最適化の目的

- 最大納期ずれの最小化

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2	5	1	3	2	0
	↓	↓	↓	↓	↓	↓	↓
M_2	1	5	3	2	3	5	6
d_j	8	10	10	7	9	16	12

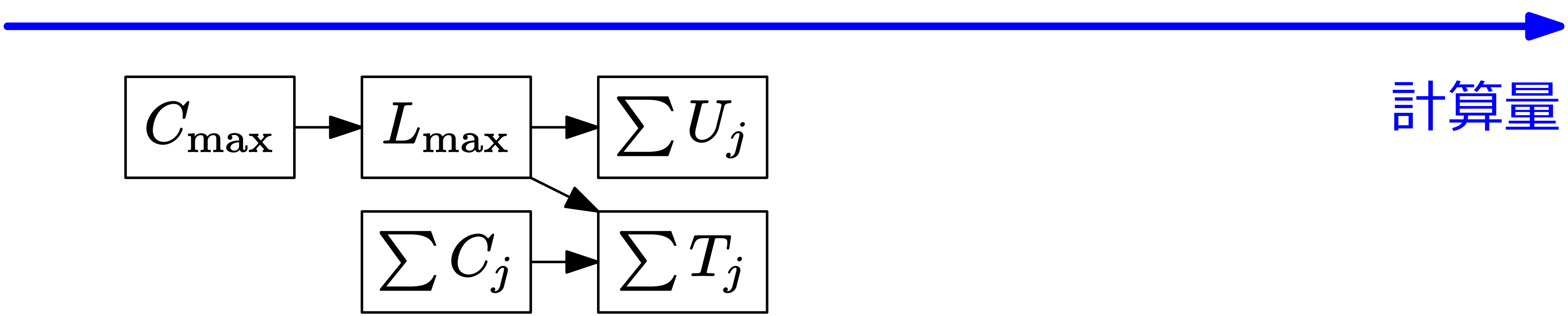
$F2 \mid\mid L_{\max}$ の強 NP 困難性

定理 (Lenstra, Rinnooy Kan, Brucker ’77)

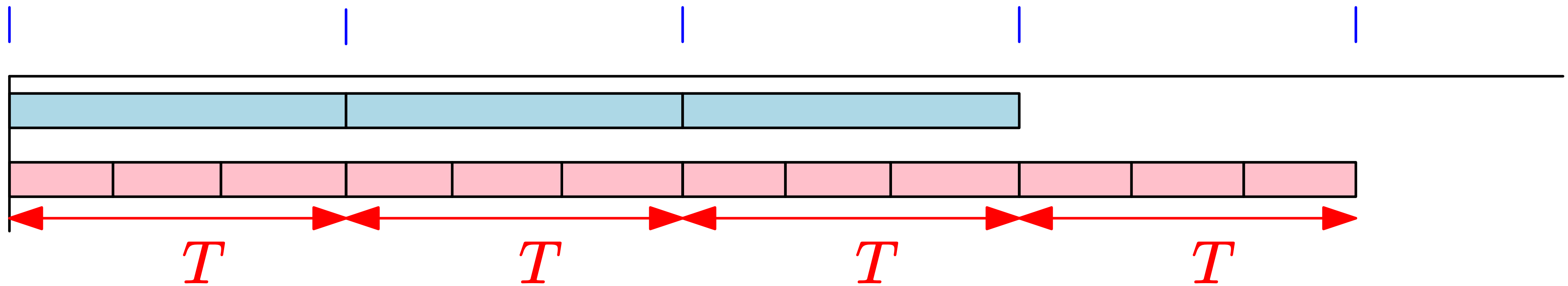
問題 $F2 \mid\mid L_{\max}$ は 強 NP 困難

注 1：$F2 \mid\mid C_{\max}$ は強多項式時間で解ける (前回)

紹介する証明は，Lenstra ら (’77) のものとは異なる



$O2 \mid\mid L_{\max}$ と同じ帰着を使う □



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	391	391	391
M_2	119	120	122	123	125	127	130	130	130	140	145	149	1	1	1	1
d_j	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1564	1	392	783	1174

今日のまとめ

- $O2 \mid\mid C_{\max}$ $O(n)$ (Gonzalez, Sahni ’76)
- $O3 \mid\mid C_{\max}$ 弱 NP 困難 (Gonzalez, Sahni ’76)

- $O2 \mid\mid L_{\max}$ 強 NP 困難 (Lawler, Lenstra, Rinnooy Kan ’81)
- $F2 \mid\mid L_{\max}$ 強 NP 困難 (Lenstra, Rinnooy Kan, Brucker ’77)

教訓

ショップ・スケジューリングは計算理論的にかなり難しい

未解決問題

$O3 \mid\mid C_{\max}$ は強 NP 困難か？